sPHENIX ladder implementation

Tony Frawley
FSU
Simulations Meeting, October 3, 2016

Overview

There are two silicon detectors planned as part of sPHENIX:

- A 3 layer MAPS inner barrel copied from the ALICE ITS upgrade
- A silicon strip detector that will act as an intermediate tracker between the MAPS inner barrel and the TPC

Both of these detectors were implemented in G4 in simplified cylinder cell geometry for the September tracking review.

We now want to adopt realistic ladder geometries in G4 for both detectors

- This is completed for the MAPS inner barrel detector
- It is completed for an obsolete design of silicon strip tracker. This should be modified to describe the intermediate tracker

Note that these have to be accommodated in the TPC simulation, since this is now our default for the sPHENIX outer tracker

MAPS ladders

MAPS ladder implementation code

The models of the three kinds of MAPS staves were created by ALICE, and they are imported whole into the sPHENIX software.

Each inner barrel layer in sPHENIX is constructed using the inner staves as the basic building block. The layer radius, azimuthal stave spacing and stave tilt are input parameters to the layer model

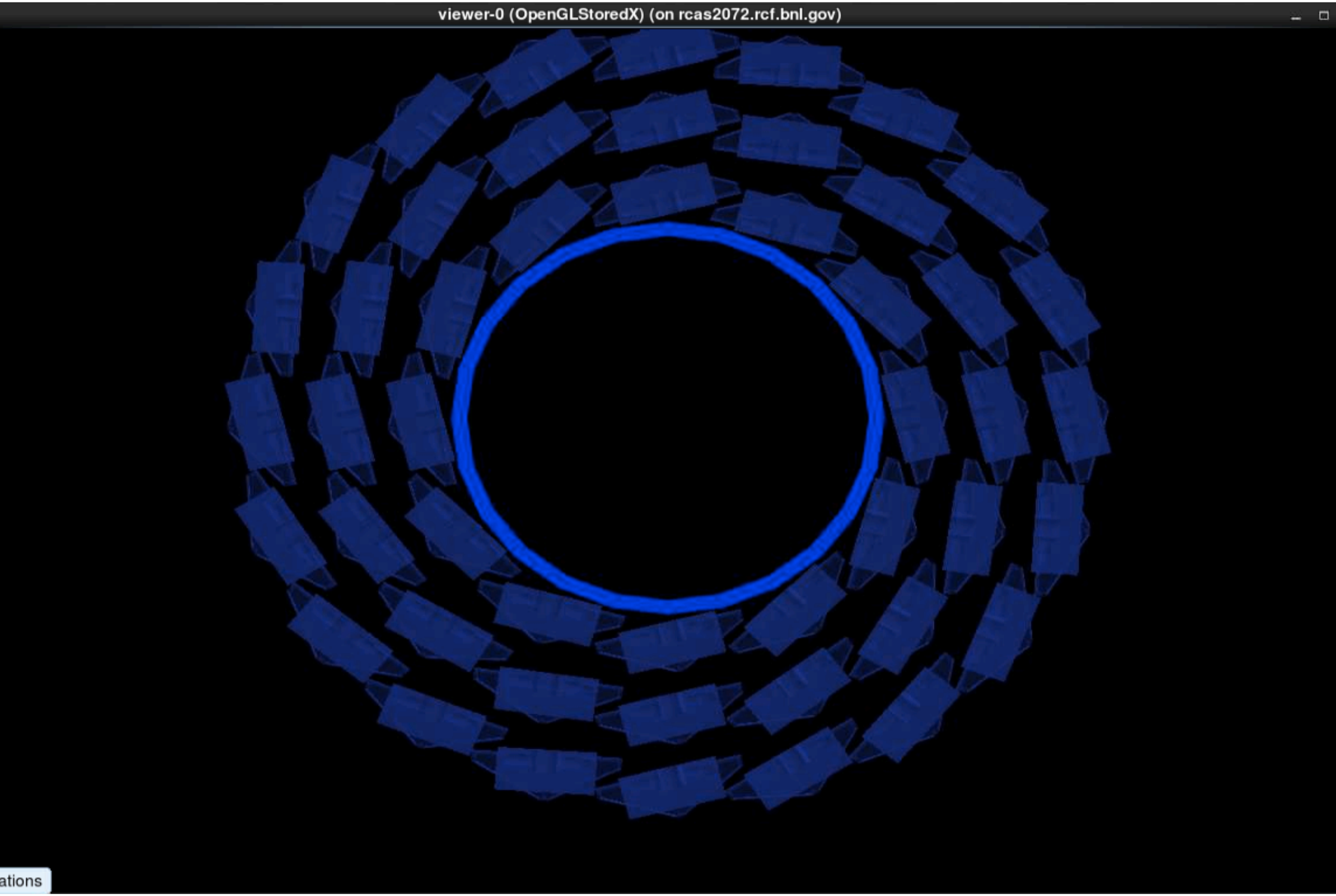
The new code that builds the model of the MAPS layers is:

PHG4MapsSubsystem	(sets parameters)
PHG4MapsDetector	(constructs barrels from the staves)
PHG4MapsSteppingAction	(collects energy in active volumes)
PHG4CylinderGeom_MAPS	(transformations between global/local)
PHG4CylinderCell_MAPS	
PHG4MapsCellReco	

The setup is created in the macro: G4_Svtx_maps_ladders+tpc.C

The ALICE geometry file is: ITS.gdml in the macros directory

The ALICE ITS staves in sPHENIX



Other code changes for MAPS ladders

Code that has been modified to accommodate the MAPS ladders implementation:

PHG4SvtxDigitizer
PHG4SvtxDeadArea
PHG4SvtxThresholds
PHG4SvtxClusterizer
PHG4TPCClusterizer
PHG4HoughTransform (used for silicon outer tracker)
PHG4HoughTransformTPC (used for TPC outer tracker - default)
SVtxEvaluator

Thresholds

The threshold settings in the standard macro:
G4_Svtx_maps+tpc.C
are set using MIP fractions of 0.25 for the cells, and 0.5 for the clusters
I found that this leads to a significant loss of efficiency for both the cylinder cell and ladders cases.

For π^+ tracks in the p_T range 0.5-8.0 GeV/c, with $-0.2 < \eta < 0.2$
(100 tracks, thrown with the same random seed)

With the cylinder cell model the track efficiency is:
78% with 0.25 mips for cell threshold and 0.5 mips for cluster threshold
95% with 0.1 mips for cell and 0.1 mips for cluster thresholds

With ladders, the track efficiency is:
56% with 0.25 mips for cell threshold and 0.5 mips for cluster threshold
91% with 0.1 mips for cell threshold and 0.1 mips for cluster threshold

Therefore I have lowered the cell and cluster thresholds to 0.1 mips

Thickness

According to the ALICE ITS model, the active sensor thickness is 18 microns. In the cylinder cell model, we are using 50 microns.

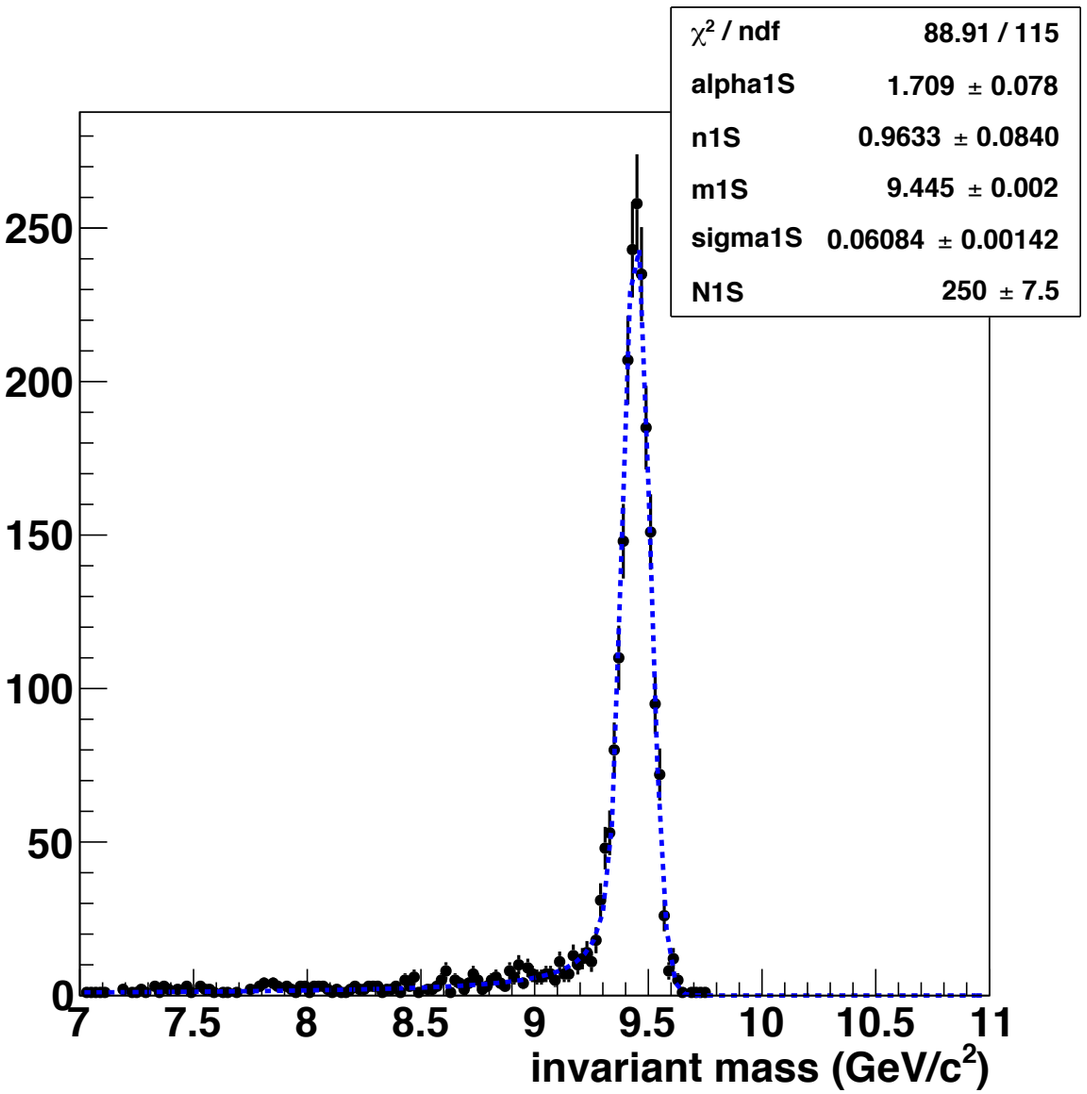
The thresholds are set using a fraction of the mip energy loss in the active thickness, so they should scale to match the thickness. However there may be secondary effects. In any case we should set the thickness in the cylinder cell model to 18 microns.

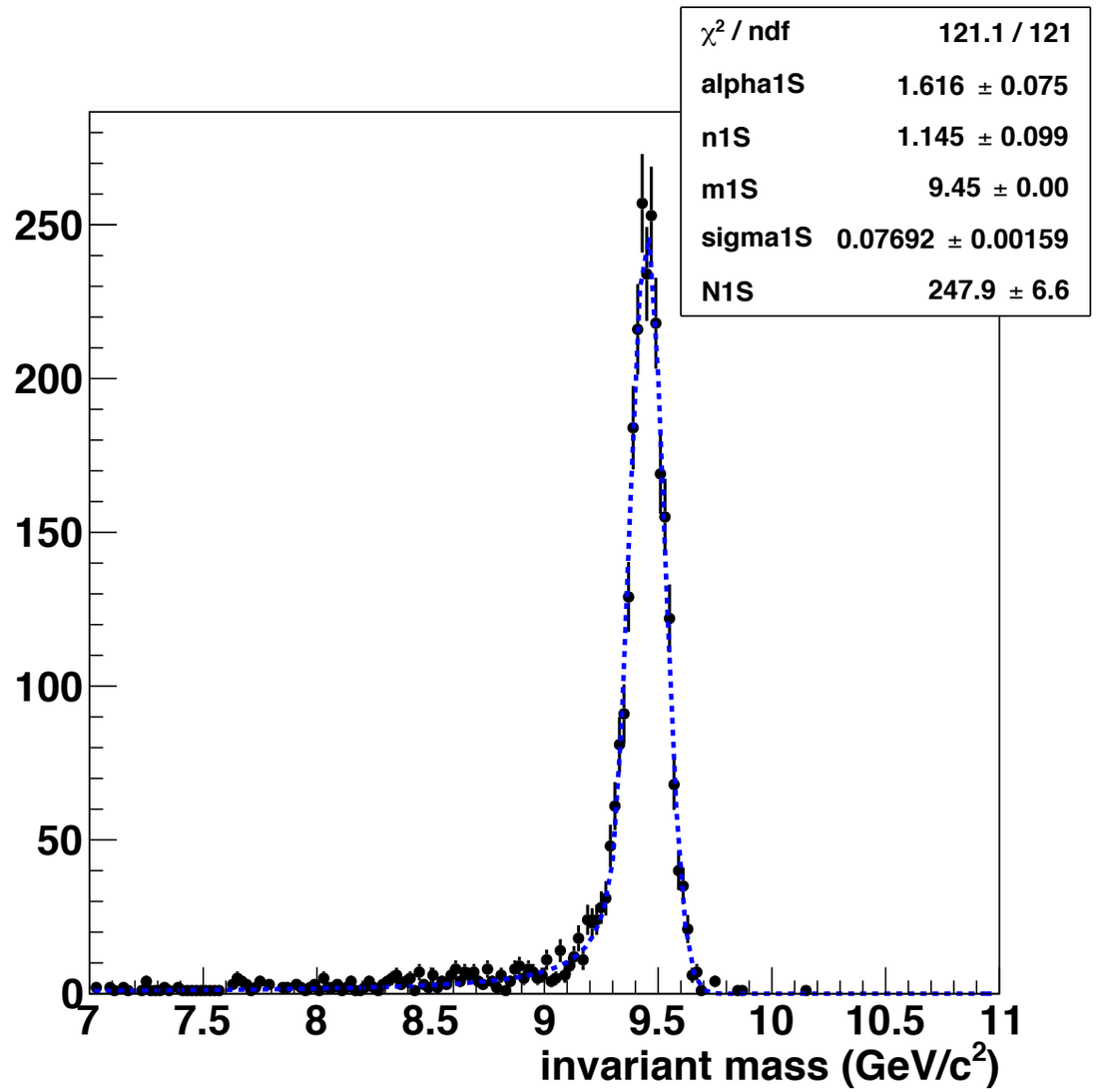
The smaller thickness leads to lower energy loss in the ladder sensors. Therefore I modified the ADC scale factor in PHG4SvtxDigitizer from 1e-06 to 0.4e-06 - this boosts the digitized ADC values back into the range used in the cylinder cell case.

There should be some more systematic tuning of thresholds

Results - Upsilon mass spectrum

Comparison of cylinder cell inner barrel and MAPS ladder inner barrel with TPC outer tracker - existing tracking code gives poorer mass resolution, higher efficiency.





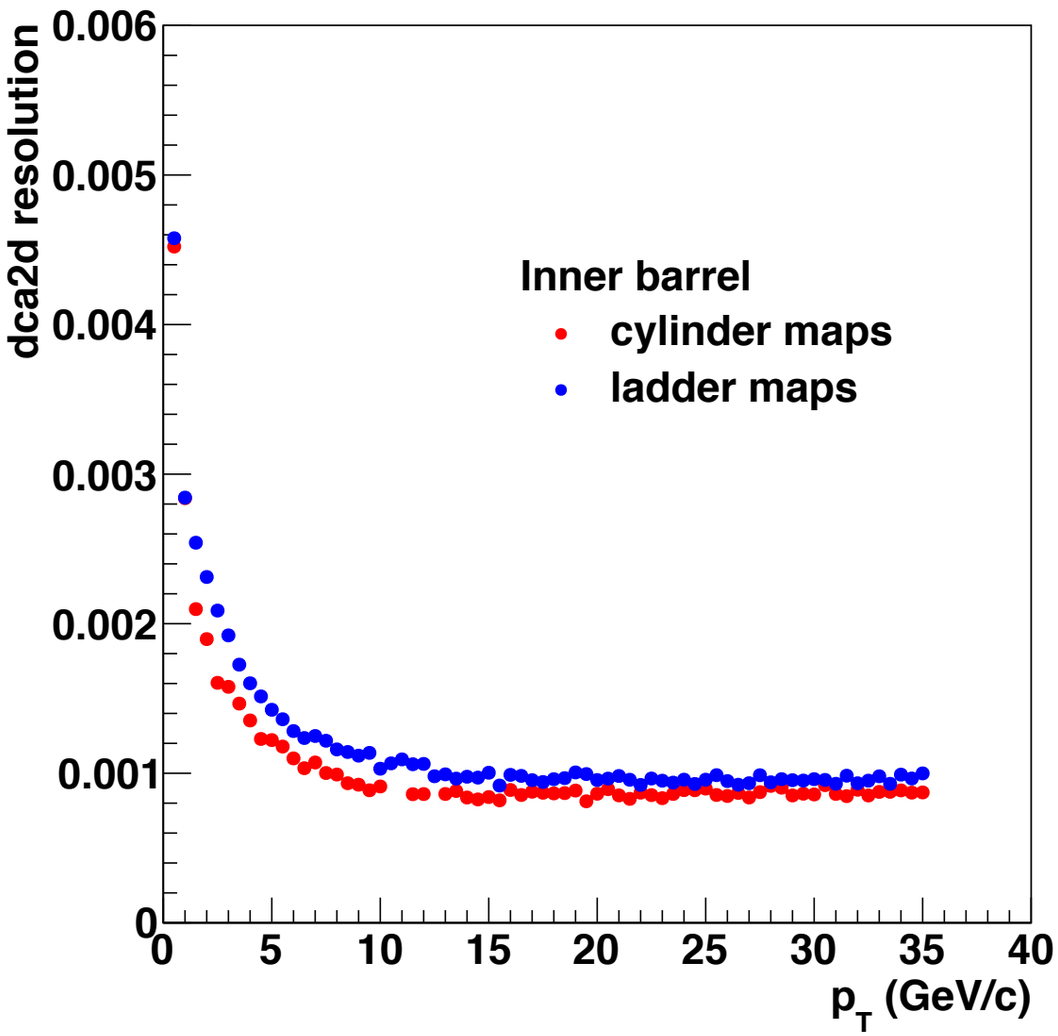
Cylinder cell
61 MeV
efficiency 39%

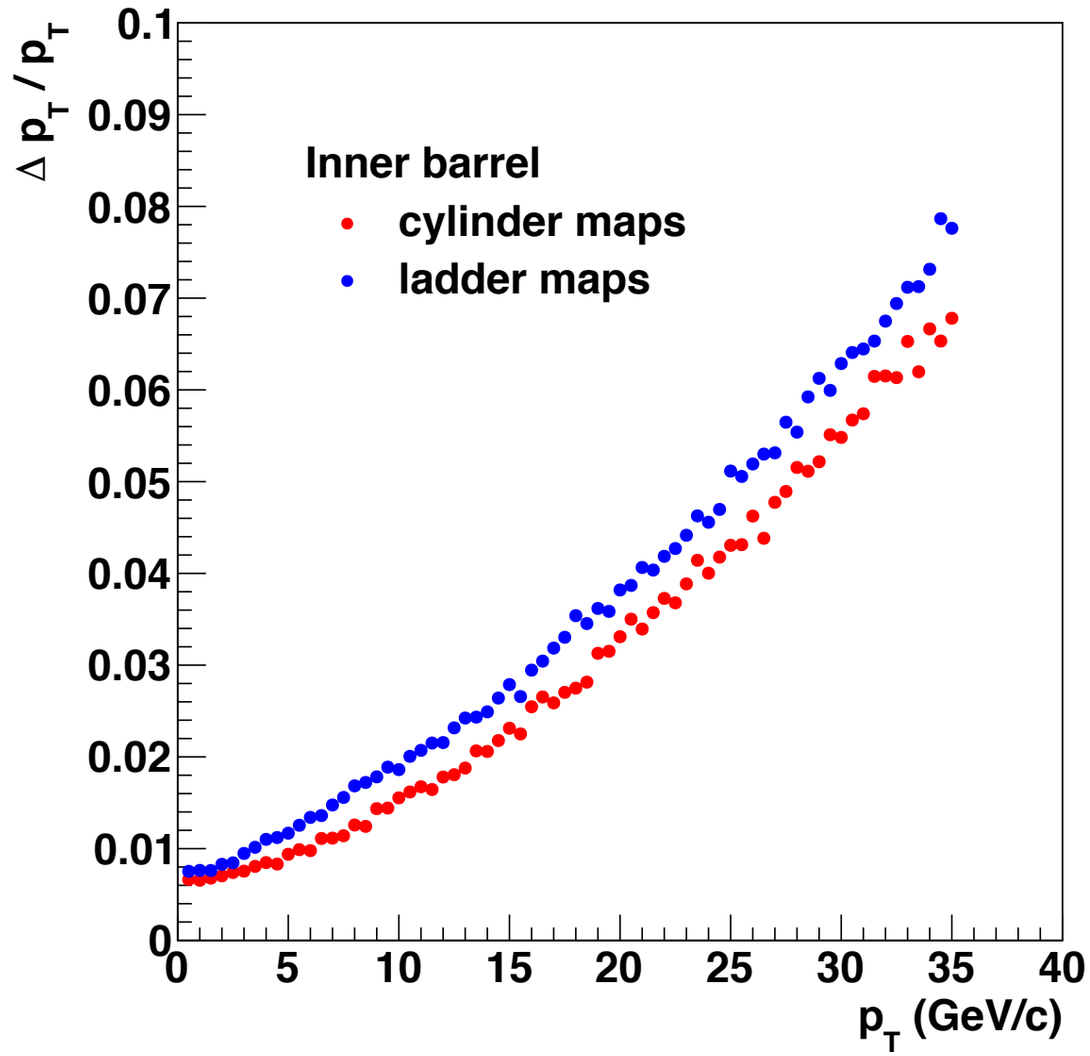
-1 < η < 1

MAPS ladders
77 MeV
efficiency 48%

Results - DCA and p_T resolution

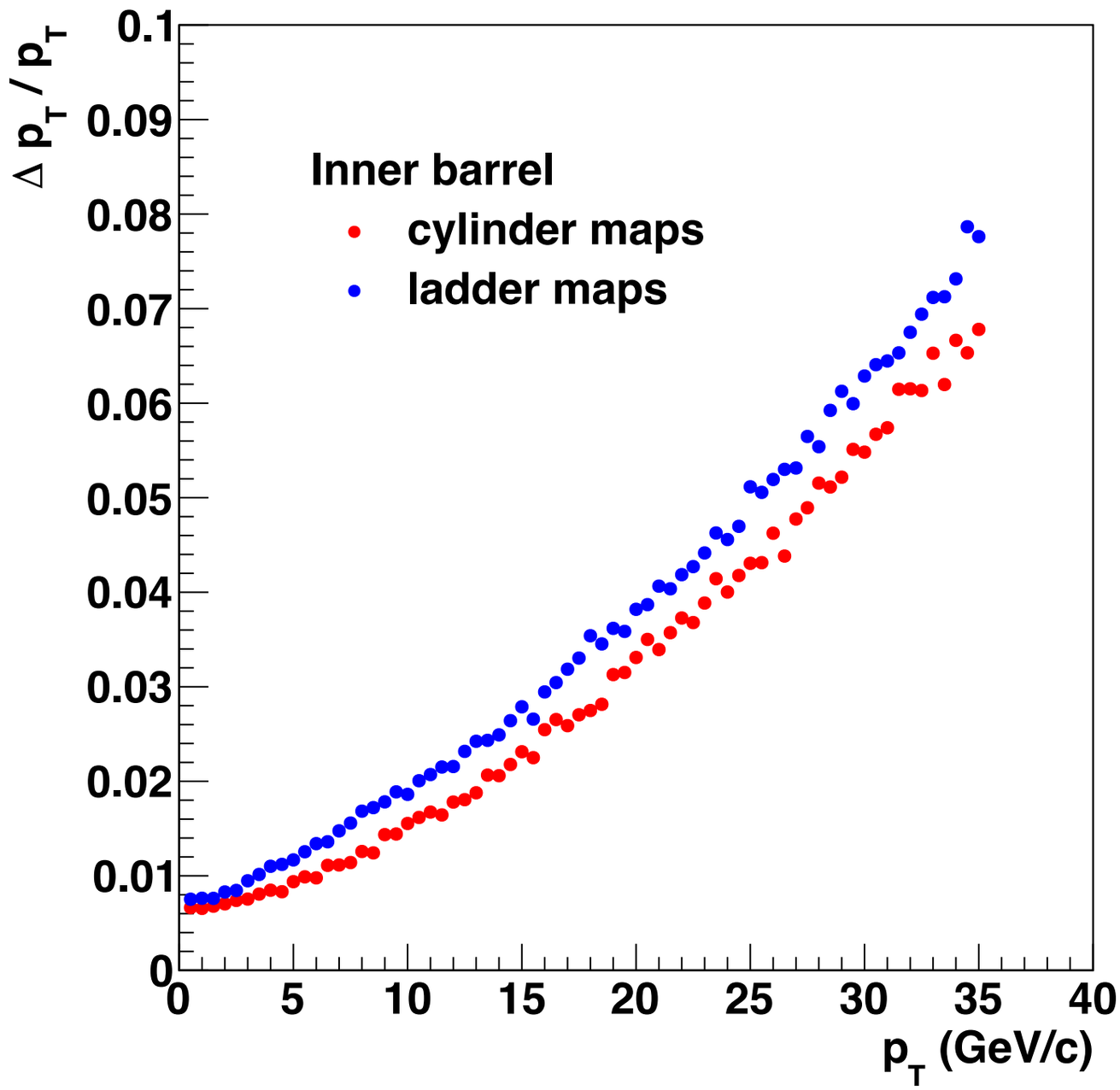
Embed 80 pions per central Hijing collision.





p_T resolution

Embedded 80 pions per central Hijing collision.



Code location for MAPS ladder implementation

The code and macros are in:

https://github.com/adfrawley/coresoftware/tree/ITS_MAPS_development/

https://github.com/adfrawley/macros/tree/ITS_MAPS_development/macros/g4simulations

These need to be merged into the main branch

Macros that are set up to produce the Upsilon spectra are in:

https://github.com/adfrawley/macros/tree/QTG_macros/macros/g4simulations

TBD for MAPS ladder implementation

The code exists only in my private branch of coresoftware

It needs to be merged into the main branch

Intermediate tracker ladders

Silicon strip ladder implementation code

There is working existing code in the main branch (written by me + Mike McCumber) that implements an obsolete design for silicon strip ladders. This should just be updated to reflect the new design.

This code builds the ladders from scratch, rather than importing the geometry, using parameters from the macro.

The existing code that builds the model of the silicon strip layers is:
PHG4SiliconTrackerSubsystem
PHG4SiliconTrackerDetector
PHG4SiliconTrackerSteppingAction
PHG4CylinderGeomv4
PHG4CylinderCellv2
PHG4SiliconTrackerCellReco

The setup is created in the macro:
G4_Svtx_ladders.C

Other code changed for silicon strip ladders

Code that was modified (by Mike McCumber) to accommodate the MAPS ladders implementation:

These changes are very similar to those that accommodate the MAPS ladders

PHG4SvtxDigitizer
PHG4SvtxDeadArea
PHG4SvtxThresholds
PHG4SvtxClusterizer
PHG4TPCClusterizer
PHG4HoughTransform (used for silicon outer tracker)
PHG4HoughTransformTPC (used for TPC outer tracker - default)
SVtxEvaluator

TBD for the intermediate tracker

What remains to be done to implement the new intermediate tracker ladders?

- Replace the algorithm in PHG4SiliconTrackerDetector that constructs the geometry
- Make any necessary adjustments to the remaining code to account for needed changes in the input parameter list, geometry object, etc….

Backup